

Beyond the Patch: Exploring Vulnerabilities of Visuomotor Policies via Viewpoint-Consistent 3D Adversarial Object

Chanmi Lee, Minsung Yoon, Woojae Kim, Sebin Lee, and Sung-eui Yoon[†]

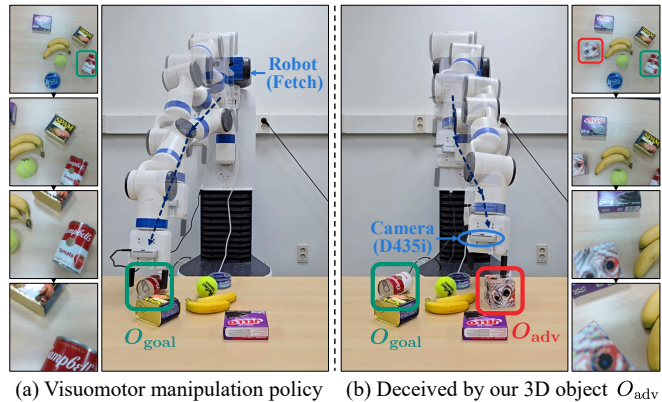
Abstract—Neural network-based visuomotor policies enable robots to perform manipulation tasks but remain susceptible to perceptual attacks. For example, conventional 2D adversarial patches are effective under fixed-camera setups, where appearance is relatively consistent; however, their efficacy often diminishes under dynamic viewpoints from moving cameras, such as wrist-mounted setups, due to perspective distortions. To proactively investigate potential vulnerabilities beyond 2D patches, this work proposes a viewpoint-consistent adversarial texture optimization method for 3D objects through differentiable rendering. As optimization strategies, we employ Expectation over Transformation (EOT) with a Coarse-to-Fine (C2F) curriculum, exploiting distance-dependent frequency characteristics to induce textures effective across varying camera-object distances. We further integrate saliency-guided perturbations to redirect policy attention and design a targeted loss that persistently drives robots toward adversarial objects. Our comprehensive experiments show that the proposed method is effective under various environmental conditions, while confirming its black-box transferability and real-world applicability.

I. INTRODUCTION

Vision-based manipulation policies have gained significant attention for enabling robots to effectively interact with objects through visual understanding [1], [2]. Among these, end-to-end approaches directly map visual inputs to actions, enabling robots to learn task-relevant features implicitly [3]–[7]. However, their reliance on neural networks makes such policies inherently vulnerable to adversarial examples, carefully crafted inputs designed to induce unintended robot behaviors [8], [9]. For instance, a malicious object unpacked from a shipment can deceive warehouse robots into hazardous actions, such as incorrect grasps or collisions.

Recent studies have begun exploring the vulnerability of visuomotor policies to adversarial attacks, primarily focusing on 2D adversarial patches [10] due to their practical feasibility [8], [9]. While 2D patches have demonstrated adversarial efficacy primarily in confined settings with fixed third-person cameras, their performance diminishes in dynamic viewpoint scenarios involving wrist-mounted cameras or mobile platforms [3]–[7], [11]. These setups induce significant viewpoint shifts, which arise not only from variations in the initial robot poses but also from continuous robot movements (Fig. 1(a)). Under such 3D viewpoint variations, the inherent planarity of 2D patches cannot fully accommodate the 3D nature of the viewpoint shifts. The planar constraint causes appearance inconsistency and severe perspective distortion at

The authors are with the School of Computing at the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, 34141, Republic of Korea. E-mails: {chanmi199, minsung.yoon, wkim97, sebin.lee}@kaist.ac.kr, sungeui@kaist.edu. [†]Corresponding author.



(a) Visuomotor manipulation policy (b) Deceived by our 3D object O_{adv}
 Fig. 1. Visuomotor policy deception using a 3D adversarial object. (a) The policy successfully guides the robot to its target O_{goal} . (b) Our adversarial object O_{adv} manipulates the visual input, compelling the policy to misguide the robot towards itself instead of the true target O_{goal} .

oblique angles, neutralizing the adversarial pattern’s effectiveness (Fig. 2). Thus, exploring viewpoint-consistent 3D adversarial objects becomes essential for evaluating security vulnerabilities and ensuring the reliability of visuomotor manipulation policies in real-world deployments.

Main Contributions. We propose a viewpoint-consistent 3D adversarial attack that effectively misleads visuomotor policies with wrist-mounted cameras. Our method optimizes a texture over a 3D mesh, leveraging Expectation over Transformation (EOT) [12] to achieve consistent attack efficacy across varied initial robot poses and continuous robot movements (Fig. 1(b)). To achieve this, we design a targeted adversarial loss that persistently misguides the robot toward the object, ensuring the adversarial object remains in the camera view throughout the task. Moreover, the attack succeeds regardless of object pose, enabling versatile placement.

To maximize attack potency, we introduce two key strategies: (1) Coarse-to-Fine (C2F) Optimization: A hierarchical strategy that ensures consistent attack efficacy across varying camera distances. We first optimize global features from distant viewpoints, then progressively refine fine-grained details from closer perspectives. (2) Saliency-Guided Attack: A strategy to redirect the policy’s focus. Using saliency maps to identify decision-critical regions, we optimize textures to shift attention from the true target to the adversarial object. Our extensive experiments demonstrate that our method not only maintains high attack efficacy under diverse viewpoint conditions but also transfers successfully to black-box scenarios and real-world applications. To the best of our knowledge, this is the first systematic analysis of visuomotor manipulation policy vulnerability to 3D adversarial attacks.

II. RELATED WORKS

A. Adversarial Examples on Robotic Manipulation Systems

Adversarial examples induce model malfunctions by adding perturbations to neural network inputs [13], [14]. In white-box attack scenarios where model architectures and parameters are fully accessible, gradient-based methods such as FGSM [14] and PGD [12] can efficiently compute optimal perturbations by leveraging the loss function’s gradients to maximize prediction errors. To enhance real-world deployability, adversarial patches localize these perturbations into printable patterns that can be physically placed in environments, creating visible yet deceptive 2D patches [10].

Recently, adversarial attacks have expanded beyond static computer vision tasks to dynamic robot systems, with a growing focus on the vulnerabilities of manipulation policies. Some approaches disrupt motion planners or Model Predictive Control (MPC) by physically manipulating the arrangement of objects in the robot’s workspace [15], [16]. By strategically configuring the environment, these methods cause the robot to misinterpret spatial layouts, leading to path-planning errors or an obstructed field of view.

In parallel, extensive research has focused on directly perturbing visual inputs to robotic policies, inducing unintended behaviors. Early works altered only one or a few pixels or applied perturbations across entire RGB images [17], [18]. Furthermore, physically deployable 2D adversarial patches placed within the robot workspace demonstrated the susceptibility of object detection models and visuomotor policies to physical adversarial attacks [8], [9]. In particular, Chen et al. [9] applied random affine transformations to enhance the physical robustness of these patches; however, fundamentally, 2D patches struggle to consistently handle viewpoint changes inherent to 3D spaces. Especially in robotic manipulation environments using wrist-mounted cameras, where viewpoint changes occur frequently due to robot arm movements [3]–[7], the effectiveness of 2D-based attacks is significantly limited, making research into viewpoint-robust attack techniques essential.

B. 3D Adversarial Examples

To develop adversarial attacks robust to diverse camera viewpoints, computer vision research has focused on various 3D adversarial attack methods optimizing textures or shapes of 3D objects [19]–[24]. These methods typically utilize Expectation over Transformation (EOT) [19], differentiable renderers [20]–[23], or Neural Radiance Fields (NeRF) [24] to ensure consistent performance across different perspectives.

Recent research in autonomous driving has also actively explored adversarial attacks considering viewpoint variations. Studies leveraging NeRF have optimized 3D object textures to effectively evaluate vulnerabilities in 3D object detection models and vision-based driving policies under varying viewpoints [25], [26]. Additionally, Chahe et al. [27] highlighted the necessity of dynamic patches capable of maintaining consistent attack despite varying distances and angles from vehicles.

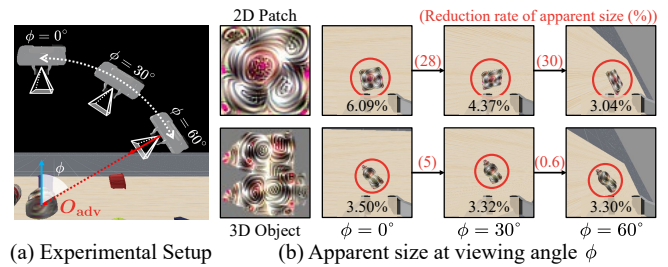


Fig. 2. Apparent size comparison of a 3D adversarial object and a 2D patch. The 2D patch significantly shrinks and distorts, especially at large viewing angles ϕ , unlike the more stable 3D object.

Ensuring viewpoint robustness is equally critical in robotic manipulation environments where frequent viewpoint changes occur. However, unlike autonomous driving, manipulation tasks face unique challenges: (1) dynamically updating patches (e.g., digital screens) is impractical in typical manipulation setups, and (2) objects are often randomly scattered on surfaces like tabletops, causing viewpoint variations and significant variations in image appearance, requiring adversarial objects to maintain effectiveness regardless of their unpredictable placement. Considering these constraints, we propose optimizing 3D adversarial object textures to robustly analyze the vulnerabilities of visuomotor manipulation policies across diverse viewpoints and random object placements.

III. METHODOLOGY

A. Problem Formulation

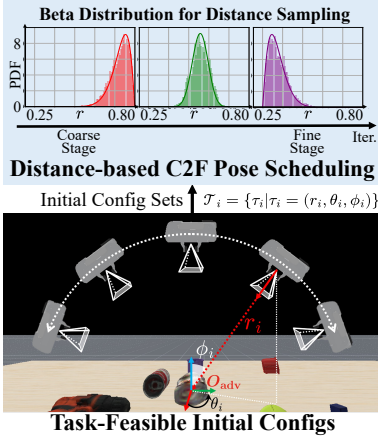
In this work, we attack an end-to-end visuomotor manipulation policy that employs a neural network π_ω . This policy maps image observations I from an eye-in-hand camera to actions $\mathbf{a} = \pi_\omega(I)$, where ω denotes the network parameters. The robot executes this policy to perform manipulation tasks (e.g., reaching a target object O_{goal}).

Against such policies, we propose a viewpoint-consistent 3D adversarial attack under a white-box scenario. We aim to craft an adversarial mesh object O_{adv} that consistently disrupts or prevents the robot from reaching the target object O_{goal} , regardless of the viewing angles encountered during task execution. To achieve this, we optimize a texture pattern for mapping onto the object’s mesh surface. This adversarial texture is designed to misguide the robot’s manipulation throughout the entire task execution by misleading the policy’s visual perception.

B. Gradient-based Texture Optimization

In this section, we present a gradient-based optimization method for adversarial object texture that maintains consistent attack effectiveness throughout manipulator movements. Our approach is driven by formulating an adversarial objective function that integrates pose alignment and model attention guidance, with optimization performed using Expectation over Transformation (EOT). Fig. 3 illustrates our overall attack framework, which consists of two main stages: Coarse-to-Fine (C2F) scheduling and the 3D adversarial attack pipeline.

(a) C2F Pose Scheduling (III.C)



(b) 3D Adversarial Object Optimization (III.B)

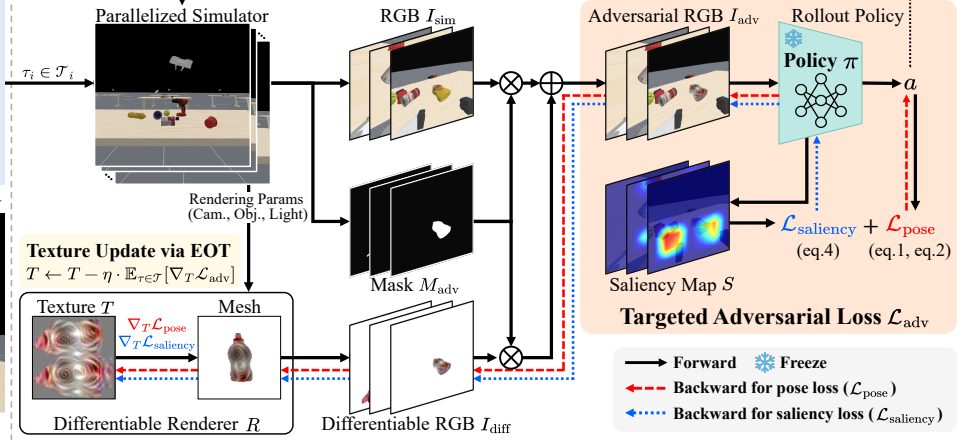


Fig. 3. Overview of the proposed method. (a) Coarse-to-Fine (C2F) Pose Scheduling: From a set of task-feasible initial configurations, poses where the original policy succeeds, we schedule viewpoint sampling using a distance-based Beta distribution. The scheduler progressively shifts focus from distant (Coarse stage) to closer (Fine stage) viewpoints. (b) 3D Adversarial Object Optimization Pipeline: Guided by the Expectation over Transformation (EOT) framework, the pipeline optimizes the adversarial texture T through short policy rollouts from each initial pose τ_i . In each step, the policy π_ω processes a composite image I_{adv} (formed from I_{diff} and I_{sim}) to output an action. A targeted adversarial loss is then computed from the resulting action, guiding the robot toward the adversarial object O_{adv} . The total loss, reflecting actual image-action pairs from the rollout, is backpropagated to update the texture T .

1) **Targeted Pose Loss:** Unlike approaches that assume static cameras, our method addresses constantly changing viewpoints induced by the robot’s own movement. In such scenarios, the adversarial object O_{adv} can fall outside the camera’s field of view throughout the robot’s movement. To maintain adversarial effectiveness throughout the entire trajectory, we design a targeted pose loss \mathcal{L}_{pose} to consistently misguide the robot’s end-effector toward O_{adv} .

The pose loss consists of two components: (1) an orientation loss \mathcal{L}_{ori} and (2) a distance loss \mathcal{L}_{dist} . The orientation loss \mathcal{L}_{ori} encourages the end-effector to point toward the adversarial object. This is achieved by maximizing the cosine similarity between the end-effector’s intended heading vector \mathbf{v}_{ee} that would result from action \mathbf{a} , and a target vector \mathbf{v}_{target} pointing from the end-effector’s action-intended position \mathbf{p}_{next} toward O_{adv} :

$$\mathcal{L}_{ori} = 1 - \frac{\mathbf{v}_{ee} \cdot \mathbf{v}_{target}}{\|\mathbf{v}_{ee}\| \|\mathbf{v}_{target}\|}. \quad (1)$$

The distance loss \mathcal{L}_{dist} minimizes the Euclidean distance between the adversarial object position \mathbf{p}_{adv} and end-effector action-intended position \mathbf{p}_{next} :

$$\mathcal{L}_{dist} = \|\mathbf{p}_{adv} - \mathbf{p}_{next}\|_2. \quad (2)$$

The total pose loss \mathcal{L}_{pose} combines both terms via the weight λ_{dist} such that $\mathcal{L}_{pose} = \mathcal{L}_{ori} + \lambda_{dist} \cdot \mathcal{L}_{dist}$.

2) **Targeted Saliency-guidance Loss:** In addition to the pose loss, we further redirect the policy’s visual attention from the original goal O_{goal} toward the adversarial object O_{adv} . We employ gradient-based saliency maps S , inspired by Grad-CAM [28], which highlight the image regions the policy focuses on to output an action. The saliency map is derived from the feature maps $A \in \mathbb{R}^{C \times H \times W}$ of the policy’s visual backbone. We compute an importance weight w_k for each channel by averaging gradients of the action norm $\|\mathbf{a}\|_2$

with respect to activations A :

$$w_k = \frac{1}{H \times W} \sum_{i,j} \frac{\partial \|\mathbf{a}\|_2}{\partial A_k(i,j)}; S = \text{ReLU} \left(\sum_k w_k A_k \right). \quad (3)$$

The saliency loss is formulated to maximize the average saliency over the adversarial object region while minimizing it over the goal object region:

$$\mathcal{L}_{saliency} = - \frac{\sum_{i,j} (S \odot M_{adv})_{i,j}}{\sum_{i,j} (M_{adv})_{i,j}} + \frac{\sum_{i,j} (S \odot M_{goal})_{i,j}}{\sum_{i,j} (M_{goal})_{i,j}}, \quad (4)$$

where M_{adv} and M_{goal} are binary masks of O_{adv} and O_{goal} , respectively.

Thus, the final adversarial loss is defined as: $\mathcal{L}_{adv} = \mathcal{L}_{pose} + \lambda_{saliency} \cdot \mathcal{L}_{saliency}$, where $\lambda_{saliency}$ balances the two terms. To ensure that the gradients from \mathcal{L}_{pose} and $\mathcal{L}_{saliency}$ do not conflict with each other during optimization, we utilize the Projecting Conflicting Gradients (PCGrad) [29] algorithm. PCGrad resolves potential conflicts by projecting the gradient of one loss onto the other and removing any opposing components before the update.

3) **Expectation over Transformation:** To ensure our adversarial object O_{adv} is effective from various viewpoints, we use the Expectation over Transformation (EOT) [12] framework. This involves optimizing the object’s texture T via gradient-based optimization, to minimize our adversarial loss \mathcal{L}_{adv} over a diverse distribution of transformations \mathcal{T} :

$$T^* = \arg \min_T \mathbb{E}_{\tau=(r,\theta,\phi) \sim \mathcal{T}} [\mathcal{L}_{adv}(T, \tau)], \quad (5)$$

where $\tau = (r, \theta, \phi)$ denotes transformations defined by distance r , azimuth angle θ , and polar angle ϕ between the adversarial object O_{adv} and the robot end-effector poses.

To ground the optimization in realism, we dynamically construct the transformation distribution \mathcal{T} by performing short rollouts that capture the robot’s actual behavior as it is influenced by the evolving adversarial texture T_t . These

rollouts originate from a set of initial configurations, where each configuration defines the initial poses for the adversarial object, the end-effector, and all other scene objects, including goals and obstacles. The scene configurations are parameterized as an initial relative pose $\tau_i = (r_i, \theta_i, \phi_i)$ between the end-effector and the adversarial object. We select only configurations where the original policy successfully completes its task, focusing on the optimization on meaningful scenarios.

The expected gradient is approximated using the collected transformations \mathcal{T} : $\mathbf{g}_t = \mathbb{E}_{(r, \theta, \phi) \sim \mathcal{T}} [\nabla_{T_t} \mathcal{L}_{\text{adv}}(T_t, r, \theta, \phi)]$, where $\nabla_{T_t} \mathcal{L}_{\text{adv}}(\cdot)$ denotes the gradient of the adversarial loss with respect to the texture at step t . We update the adversarial texture T using the expected gradient as: $T_{t+1} = \text{clip}(T_t - \eta \cdot \mathbf{g}_t / \|\mathbf{g}_t\|_2, 0, 1)$, where η is the learning rate. The texture values are clipped to the valid range $[0, 1]$. This iterative optimization yields an adversarial texture maintaining attack efficacy across variations in camera viewpoints.

4) **Differentiable Rendering**: Optimizing the adversarial texture T requires the gradient of the adversarial loss \mathcal{L}_{adv} with respect to the texture. However, standard robot simulators rely on non-differentiable operations, such as rasterization, making gradient computation challenging [21], [23]. To mitigate this, we employ a hybrid rendering strategy to bypass the simulator’s non-differentiable components during adversarial object optimization. Specifically, the overall scene is rendered with the standard simulator, while the adversarial object O_{adv} is rendered separately using a differentiable renderer R . The final composed image I_{adv} is defined as: $I_{\text{adv}} = (1 - M_{\text{adv}}) \odot I_{\text{sim}} + M_{\text{adv}} \odot I_{\text{diff}}$, where I_{sim} and I_{diff} denote images from the standard and differentiable renderers, respectively, M_{adv} is a binary mask for the adversarial object, and \odot denotes the Hadamard product. This hybrid approach enables gradient computation with respect to the adversarial texture, enabling iterative texture optimization.

C. Coarse-to-Fine Attack Strategy

In robotic manipulation with wrist-mounted cameras, the camera-object distance varies constantly and significantly. While adversarial textures must remain effective across all viewing distances, simultaneous optimization for multiple distances often leads to conflicting objectives that degrade overall attack performance [30]. To this end, we propose a distance-based Coarse-to-Fine (C2F) optimization strategy. Our approach leverages a key observation that the optimizable texture features depend on viewing distance due to changes in apparent resolution. At longer distances, primarily low-frequency (coarse) features remain distinguishable, while at shorter distances, high-frequency (fine) details can be effectively optimized.

Reflecting this distance dependence, we adopt a Coarse-to-Fine (C2F) optimization strategy, optimizing sequentially from far-range to near-range scenarios. In the initial **Coarse Stage**, we first optimize low-frequency components at longer viewing distances to establish robust global texture patterns. Building upon this foundation, the **Fine Stage** then refines high-frequency details crucial for near-distance effectiveness. By prioritizing coarse global consistency before fine-level

refinements, the C2F strategy is better suited for generating adversarial textures effective throughout the robot’s manipulation trajectory.

We implement this C2F progression within the EOT framework (Sec. III-B.3) by scheduling the sampling of initial configurations $\tau_i \in \mathcal{T}_i$ based on their distance r_i . For each optimization stage, we sample configurations with specific camera-object distances r_i according to a scheduled distribution. To ensure smooth transitions between stages while maintaining focus on target distance ranges, we employ a Beta distribution to control sampling probability. Throughout the optimization process, we progressively adjust the Beta parameters (α, β) of the Beta distribution to systematically shift the sampling priority from longer distances (Coarse Stage) to shorter ones (Fine Stage).

As a result, the model first learns coarse patterns that provide a stable foundation, then refines the fine details necessary for close-range effectiveness.

IV. EXPERIMENTAL RESULT

A. Experimental Setup

1) **Simulation and Task**: We conduct experiments in the SAPIEN-based ManiSkill3 framework [31], [32] using a floating Panda gripper with a wrist-mounted camera. The target is an end-to-end visuomotor policy π_ω with a ResNet18 backbone [33], trained via PPO [34] to reach a target object O_{goal} . For this task, we define O_{goal} as a ‘tomato_soup.can’ and O_{adv} as a ‘mustard_bottle’ from the YCB dataset [35].

2) **Adversarial Texture Optimization and C2F Staging**: We optimize the adversarial texture for 10k iterations using online data from 10-step rollouts across 4 environments, starting with the end-effector 25–80 cm from O_{adv} . For the optimization process, we set hyperparameters as follows: $\eta=0.1$, $\lambda_{\text{dist}}=0.1$, and $\lambda_{\text{saliency}}=0.01$. The Coarse-to-Fine (C2F) strategy proceeds in five sequential stages, implemented via Beta scheduling with stage-specific (α, β) as (13.48, 2.39), (23.13, 10.49), (19.88, 19.88), (10.49, 23.13), and (2.39, 13.48), progressing from coarse to fine granularity. All evaluations employ the standard SAPIEN renderer.

3) **Computational Resources**: Offline texture optimization takes approximately 5 hours on an RTX 4090. Each of the 10k iterations takes about 1.5 seconds and 2.5 GB of VRAM.

4) **Evaluation Metrics**: To accurately assess attack performance, we ran 500 complete 60-step episodes for each setup, starting from task-feasible initial configurations where the baseline policy succeeds. From these episodes, we computed task-level success via ASR and T-ASR, and momentary action error via $\mathcal{E}_{\text{trans}}$, \mathcal{E}_{rot} . Higher ASR, T-ASR, $\mathcal{E}_{\text{trans}}$, and \mathcal{E}_{rot} indicate a stronger attack. All results represent 3-seed means with ASR and T-ASR standard deviations below 2%.

- **Attack Success Rate (ASR)**: Failure rate to reach O_{goal} .
- **Targeted Attack Success Rate (T-ASR)**: Success rate of redirecting the robot toward O_{adv} .
- **Translation Error ($\mathcal{E}_{\text{trans}}$)**: Action translation deviation between original and attacked observations.
- **Rotation Error (\mathcal{E}_{rot})**: Action rotation deviation between original and attacked observations.

TABLE I. Attack Performance (%) Comparison: Adversarial 3D Object (Ours) vs. 2D Patch Attack Across Viewing Angles (ϕ°)

Attack Type	Results by Viewing Angle (ϕ°)																Avg	
	0-10		10-20		20-30		30-40		40-50		50-60		60-70		70-90		T-ASR	ASR
	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR	T-ASR	ASR
2D Patch	73.00	77.00	61.40	74.40	54.80	66.80	46.40	65.00	31.00	56.60	27.00	51.80	15.00	48.80	10.60	44.20	39.90	60.58
3D Object	78.00	79.00	74.00	80.00	62.00	77.00	64.00	74.00	60.00	70.00	51.00	62.00	40.00	55.00	34.20	56.60	57.90	69.20

TABLE II. Attack Performance Comparison (T-ASR(\uparrow), ASR(\uparrow) in %; $\mathcal{E}_{\text{trans}}(\uparrow)$, $\mathcal{E}_{\text{rot}}(\uparrow)$ Action Errors) Across Optimization Strategies (C2F, F2C, NS) with Ablations on Auxiliary Losses (Saliency, Targeted, Pose) at Varying Initial Robot Distances (cm).

	Opt. Strategy	Adversarial Losses			Results by Distance (cm)										Avg	
		Targeted	Pose	Saliency	25-36		36-47		47-58		58-69		69-80		T-ASR	ASR
					$\mathcal{E}_{\text{trans}}$	\mathcal{E}_{rot}	T-ASR	ASR	$\mathcal{E}_{\text{trans}}$	\mathcal{E}_{rot}	T-ASR	ASR	$\mathcal{E}_{\text{trans}}$	\mathcal{E}_{rot}	T-ASR	ASR
(a)	NS	✓	✓	✓	56.00	59.60	69.20	70.40	73.80	75.20	75.00	76.80	94.60	95.20	73.72	75.44
					0.33	0.042	0.34	0.040	0.31	0.036	0.29	0.034	0.26	0.033	0.31	0.037
(b)	F2C	✓	✓	✓	54.40	55.80	59.20	61.60	67.40	70.60	64.80	67.40	89.40	91.40	67.04	69.36
					0.32	0.034	0.32	0.034	0.30	0.032	0.28	0.030	0.26	0.029	0.30	0.032
(c)	C2F		✓	✓	0.20	15.00	0.00	8.00	00.00	10.20	00.00	8.40	0.00	34.40	0.04	15.20
					0.05	0.006	0.04	0.005	0.03	0.005	0.03	0.004	0.02	0.004	0.04	0.005
(d)	C2F	✓		✓	60.60	63.80	72.20	73.60	73.80	75.60	72.60	73.60	89.80	91.40	73.80	75.60
					0.21	0.030	0.22	0.032	0.24	0.035	0.24	0.035	0.26	0.038	0.23	0.034
(e)	C2F	✓	✓		59.20	62.60	73.80	74.20	78.60	79.80	82.00	82.80	96.80	97.40	78.08	79.36
					0.49	0.061	0.48	0.057	0.43	0.050	0.37	0.045	0.33	0.041	0.42	0.051
(f)	C2F	✓	✓	✓	61.60	65.60	76.60	77.00	82.20	83.40	84.20	85.20	97.40	97.60	80.36	81.76
					0.53	0.066	0.51	0.060	0.45	0.054	0.40	0.048	0.35	0.044	0.45	0.055

B. Result Analysis

Effectiveness of 3D Attack: To evaluate robustness to viewpoint changes, we compared the proposed 3D attack with a 2D patch-based attack. We designed a fair 2D baseline to isolate performance differences caused purely by dimensionality. This was achieved by optimizing a thin cuboid’s top surface within the 3D attack’s rendering and optimization environment, which avoided traditional transformations [22] and kept all other conditions identical.

As shown in TABLE I, our 3D attack consistently achieved higher ASR and T-ASR. The performance gap widened at oblique angles (large ϕ), with the 3D attack’s T-ASR being over twice as high beyond 60° . This is because a 2D patch’s projected area shrinks and distorts from such viewpoints, whereas the 3D object maintains a larger, more stable projection. This demonstrates the greater robustness of 3D texture optimization to viewpoint changes, making it essential for applications like eye-in-hand robotics.

C. Ablation Studies

1) Effect of Coarse-to-Fine Optimization Strategy: To validate our proposed Coarse-to-Fine (C2F) attack strategy, we compared its performance against several alternative optimization strategies: (1) Non-staged (uniform sampling across the distance range without staging), (2) Fine-to-Coarse (F2C) (reverse C2F order), (3) Coarse-only, and (4) Fine-only. Fig. 4 and TABLE II present their visual and quantitative comparisons, respectively. The **Coarse-only** method (Fig. 4 (d)) produces simple textures based on low-frequency coarse features, while the **Fine-only** method (Fig. 4(e)) generates intricate patterns rich in high-frequency fine details. Our **C2F** method (Fig. 4 (a)) creates a well-balanced texture by effectively building fine details upon a stable coarse

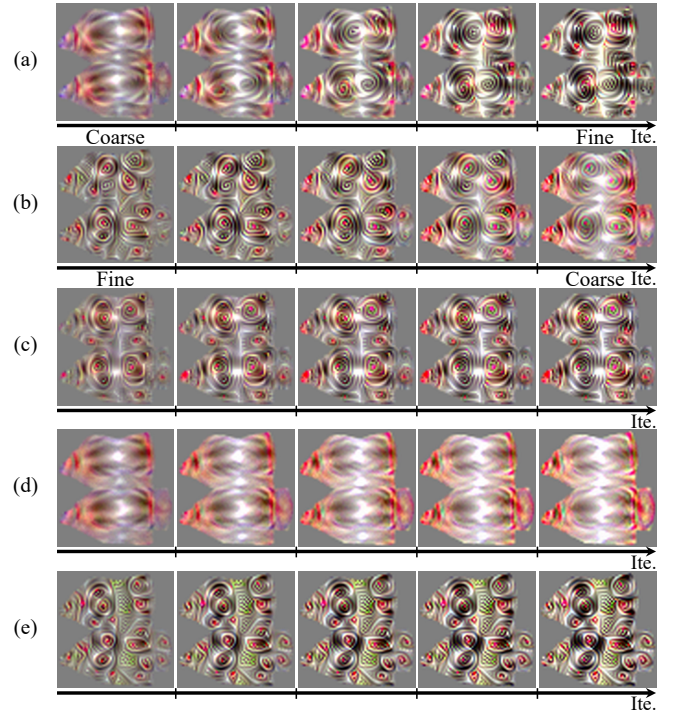


Fig. 4. Visualization of texture update patterns under different scheduling strategies: (a) Coarse-to-Fine, (b) Fine-to-Coarse, (c) Non-staged, (d) Coarse-only, (e) Fine-only.

foundation. Conversely, the **F2C** method (Fig. 4(b)) shows that initially learned fine details become blurred during the subsequent coarse stage, indicating inefficiency in the optimization path. The **Non-staged** approach (Fig. 4(c)), which uniformly samples from all distances, results in mixed fine and coarse features throughout the optimization process. Quantitatively, as shown in TABLE II, the proposed C2F method not only achieved higher attack success rates (ASR

and T-ASR) but also exhibited higher $\mathcal{E}_{\text{trans}}$ and \mathcal{E}_{rot} values compared to the other scheduling methods. These results confirm that the C2F strategy, establishing global structure before refining details, is crucial for optimizing adversarial textures that maintain consistent effectiveness across the dynamic viewing conditions of wrist-mounted cameras, especially under distance variations. By building a solid coarse foundation first and then adding fine details, our approach ensures reliable attack performance despite the continuous changes in viewpoint and distance inherent to wrist camera movements during manipulation tasks.

2) **Effect of Saliency Guidance:** This section analyzes the effect of saliency guidance, which improves optimization efficiency by focusing the attack on regions deemed important by the target policy π_{ω} . We evaluated its impact by comparing C2F performance with and without saliency guidance, as shown in TABLE II (d)-(f).

Comparing TABLE II (e) and (f), we observe that incorporating saliency loss improves the attack performance across all metrics (ASR, T-ASR, $\mathcal{E}_{\text{trans}}$, \mathcal{E}_{rot}). Additionally, the results in TABLE II (d) demonstrate that using targeted saliency loss alone can achieve reasonable attack performance, confirming its individual contribution to the overall effectiveness.

The effect of saliency guidance can be observed in the attention shift illustrated in Fig. 5. Before the attack (Fig. 5 (a)), the policy’s attention focuses on the goal object (O_{goal}). After the attack (Fig. 5 (b)), attention partially shifts toward the adversarial object (O_{adv}). This result visually demonstrates that our attack improves performance by successfully controlling the policy’s attention.

3) **Effect of Targeted Loss:** To handle dynamic camera motion and continuously changing viewpoints, an adversarial objective must not only disrupt the policy but also keep the adversarial object O_{adv} within the camera’s field of view (FOV). We evaluate our targeted loss \mathcal{L}_{adv} (Sec. III-B) against an untargeted loss that simply disrupts the original goal-oriented behavior.

The untargeted loss disrupts goal-oriented behavior by maximizing the action divergence between an adversarial image (I_{adv}) and a normal one (I_{sim}), while also reducing visual attention toward the goal object (O_{goal}):

$$\mathcal{L}_{\text{untargeted}} = -\|\mathbf{a} - \mathbf{a}_{\text{gt}}\|^2 + \lambda_{\text{saliency}} \cdot \frac{\sum (S \odot M_{\text{goal}})}{\sum M_{\text{goal}}}. \quad (6)$$

Here, $\mathbf{a} = \pi_{\omega}(I_{\text{adv}})$ is the action on the adversarial image and $\mathbf{a}_{\text{gt}} = \pi_{\omega}(I_{\text{sim}})$ is the ground-truth action on the normal one. The loss maximizes the Mean Squared Error (MSE)

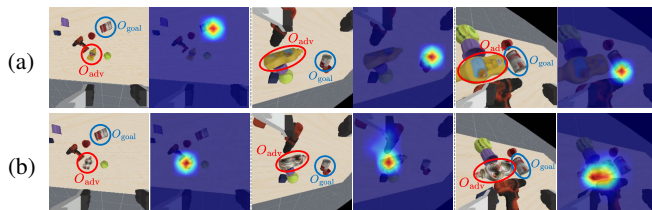


Fig. 5. Comparison of policy saliency maps: (a) before vs. (b) after the 3D adversarial attack.

between these actions to force behavioral deviation, while the saliency term penalizes attention on the goal mask (M_{goal}) to divert the policy’s focus.

As shown in TABLE II (f), our targeted loss \mathcal{L}_{adv} yields higher ASR and T-ASR, demonstrating stronger robustness to viewpoint changes. In contrast, the untargeted baseline (TABLE II (c)) shows inferior performance. Due to a lack of explicit guidance toward O_{adv} , the attack causes only intermittent disruptions and allows the policy to recover once the adversarial object O_{adv} exits the field of view. These results indicate that \mathcal{L}_{adv} maximizes task failure by ensuring the visibility of O_{adv} through persistent guidance of the robot.

D. Generalization and Robustness Analysis

1) **Generalization to Diverse Object Geometries:** To verify the generalization performance of our proposed attack, we experimented with objects of various geometric structures, such as dog and duck shapes, as they are everyday objects with distinctly different and complex morphologies. This result confirms that our method is not overfitted to a specific geometry, demonstrating its ability to generate effective adversarial features for any given morphology.

2) **Transferability to Different Camera Configurations:** We evaluated the transferability of our single wrist-camera attack on a stereo camera setup. The attack transferred effectively, maintaining high T-ASR and ASR, which demonstrates its robustness against significant changes in camera configuration.

3) **Robustness to Environmental Variations:** We evaluated the attack’s robustness against visual environmental changes, including lighting (Light:Bright, Light:Dim), background (Bkg Varied), and sensor noise (Add Noise). The experimental results showed only minimal performance degradation, demonstrating that the proposed method operates stably even in environments with realistic variations. The results for Sec. IV-D are summarized in TABLE III.

Note: Detailed experimental settings and results for this Sec. IV-D are available in the supplementary video.

E. Validation in Realistic Scenarios

1) **Transferability to Black-Box Scenario:** We further evaluate our method on a more practical black-box scenario where adversaries lack knowledge about the architecture of the target policy. To do so, we measure the ability of

TABLE III. Attack Generalization and Robustness Analysis (%)

Sec.	Condition	T-ASR	ASR	Sec.	Condition	T-ASR	ASR
	Ours	79.80	81.12		Light: Bright	77.04	84.70
D.1	Shape: Dog	60.04	65.80	D.3	Light: Dim	67.80	72.20
	Shape: Duck	63.20	68.32		Add Noise	75.57	80.41
D.2	Stereo Cam	63.28	78.91		Bkg Varied	77.91	85.64

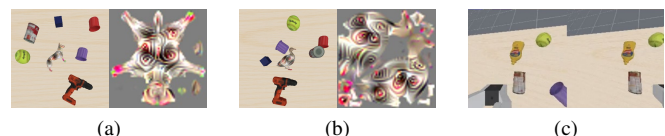


Fig. 6. Visualization of Sec. IV-D configurations: (a) D.1. Dog shape, (b) D.1. Duck shape, (c) D.2. Stereo camera setup.

TABLE IV. Attack Transferability Across Visual Policy Architectures (Source: ResNet18[†], (%))

Policy	T-ASR	ASR	Policy	T-ASR	ASR
ResNet18 [†]	79.08	81.12	Inception-v3	56.68	70.28
VGG16	59.68	70.08	ResNet34	71.20	78.32

our attack optimized on a white-box source model using ResNet18 with PPO to fool unseen black-box target models using Inception-v3, VGG16, or ResNet34. As shown in TABLE IV, the attack success rates remain relatively high across these diverse architectures. This demonstrates that our method generalizes beyond the specific source architecture and poses a viable threat in realistic black-box scenarios.

2) *Sim-to-Real Transferability*: We evaluate the direct transferability of our simulation-generated adversarial objects to real-world environments. To mitigate the sim-to-real gap, we incorporated lighting domain randomization during texture optimization. The experiments were performed with a Fetch robot and a wrist-mounted RealSense D435i camera in two settings: EnvA, where the policy aims to reach a YCB [35] ‘tomato_soup.can’, and EnvB, where the target is a cube-shaped object (see Fig. 8 (b),(c)). We test two 3D objects (a cube and a cylinder) for 30 trials each, covering 5 initial end-effector poses and 6 object poses where the baseline policy succeeds. A reach attempt was considered successful if the gripper moved to within 10cm of the target

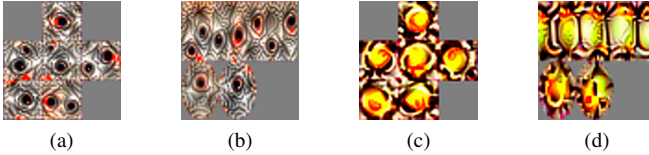


Fig. 7. Visualization of adversarial textures on real-world objects: (a) Cube in EnvA; (b) Cylinder in EnvA; (c) Cube in EnvB; (d) Cylinder in EnvB.

TABLE V. Attack Sim-to-Real Transferability Evaluation (%)

Env	Shape of O_{adv}	Simulation		Real-World		Note
		T-ASR	ASR	T-ASR	ASR	
A	Cube	71.60	80.60	60.00	73.33	Fig. 1(b)
A	Cylinder	76.20	84.60	73.33	76.67	Fig. 8(a)
B	Cube	72.80	83.00	50.00	60.00	Fig. 8(b)
B	Cylinder	75.40	85.40	53.33	63.33	Fig. 8(c)



Fig. 8. Policy rollout with (a) a cylinder-shaped adversarial object in EnvA, (b) a cube-shaped adversarial object in EnvB, and (c) a cylinder-shaped adversarial object in EnvB.

(adversarial object O_{adv} or goal object O_{goal}), and this target was the closest object to the gripper.

The results, shown in Fig. 8 and TABLE V, confirm that our attack is effective in the real world. Despite a slight performance degradation due to sim-to-real gaps like lighting, shadows, and printing quality, the adversarial objects successfully misled the policy.

3) *Robustness in Challenging Scenarios*: To further assess the attack’s resilience, we evaluate its effectiveness in more complex scenarios involving dynamically moving and partially occluded adversarial objects. For the occlusion experiments, we test scenarios where 40-70% of the adversarial object is obscured by other objects or obstacles. As depicted in Fig. 9 and TABLE VI, our attack maintains effectiveness even when object positions change mid-task or when objects are substantially occluded, demonstrating applicability beyond simple static settings.

TABLE VI. Attack Robustness (%) Under Challenging Scenarios

Env	Scenario	T-ASR	ASR	Env	Scenario	T-ASR	ASR
A	Dynamic object (a)	46.67	60.00	A	Occluded scene (b)	43.33	56.67
B	Dynamic object	40.00	50.00	B	Occluded scene	36.67	50.00

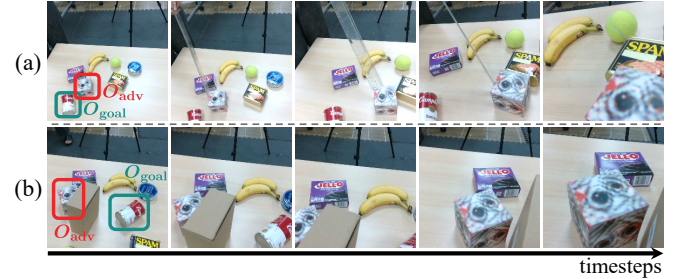


Fig. 9. Policy rollout in EnvA: (a) with the dynamically moved adversarial object; (b) with the partially occluded adversarial object.

TABLE VII. Attack Robustness (%) Under Different Lighting Variations

Env	Lighting	T-ASR	ASR	Env	Lighting	T-ASR	ASR
A	Dim1	50.33	66.67	A	Bright1	53.33	73.33
A	Dim2	53.33	66.67	A	Bright2	56.67	73.33
A	Dim3	53.33	70.00	A	Dynamic	36.67	50.00
B	Dim1	46.67	60.00	B	Bright1	50.00	66.67
B	Dim2	50.00	60.00	B	Bright2	53.33	66.67
B	Dim3	50.00	63.33	B	Dynamic	33.33	50.00

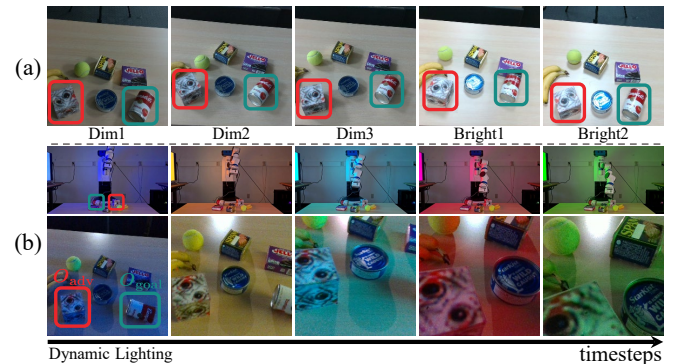


Fig. 10. (a) Setups for assessing robustness to lighting in EnvA (from left: dim1, dim2, dim3, bright1, and bright2); (b) policy rollout in dynamic lighting projected via an LED monitor.

4) **Robustness to Environmental Variations:** We assess attack performance under diverse real-world lighting conditions to evaluate robustness in practical deployment settings. The results presented in Fig. 10 and TABLE VII demonstrate that the attack maintains consistent effectiveness despite substantial illumination variations (dim, bright, and dynamic lighting) and cluttered or varying backgrounds.

Note: Additional experimental configurations of EnvB for Sec. IV-E.3 and Sec. IV-E.4 are demonstrated in the supplementary video.

V. CONCLUSION

In this paper, we propose a viewpoint-consistent 3D adversarial attack method that effectively disrupts visuomotor policies under dynamic camera viewpoints. Our method achieves this by extending 2D patches into optimized 3D objects, incorporating two key innovations: a Coarse-to-Fine (C2F) optimization strategy to ensure robustness against distance variations, and a saliency-based approach that enhances attack efficiency by targeting critical visual areas. Extensive experiments demonstrate not only superior performance compared to 2D patch attacks but also strong generalization across object geometries, camera setups, and black-box models. Furthermore, our results validate the sim-to-real transferability of adversarial objects, underscoring the real-world threat posed to robotic systems. Ultimately, our work provides both a novel attack methodology and a practical evaluation tool for strengthening the reliability of robot perception and control in safety-critical applications.

ACKNOWLEDGMENT

This work was supported by the Korea government (MSIT) through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (RS-2025-25443318) and the National Research Foundation of Korea (NRF) grant (RS-2023-00208506).

REFERENCES

- [1] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Object-centric imitation learning for vision-based robot manipulation," in *CoRL*, 2022.
- [2] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *ICRA*, 2018.
- [3] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.
- [4] M. Seo, H. A. Park, S. Yuan, Y. Zhu, and L. Sentis, "Legato: Cross-embodiment imitation using a grasping tool," *IEEE Robotics and Automation Letters*, 2025.
- [5] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.
- [6] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation," in *CoRL*, 2024.
- [7] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "Data scaling laws in imitation learning for robotic manipulation," in *ICLR*, 2025.
- [8] Y. Jia, C. M. Poskitt, J. Sun, and S. Chattopadhyay, "Physical adversarial attack on a robotic arm," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9334–9341, 2022.
- [9] Y. Chen, H. Xue, and Y. Chen, "Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies," in *NeurIPS*, 2024.
- [10] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.
- [11] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "Okami: Teaching humanoid robots manipulation skills through single video imitation," in *CoRL*, 2024.
- [12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.
- [13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.
- [14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
- [15] W. Wu, F. Pierazzi, Y. Du, and M. Brandão, "Characterizing physical adversarial attacks on robot motion planners," in *ICRA*, 2024.
- [16] S. Agarwal and S. P. Chinchali, "Synthesizing adversarial visual scenarios for model-based robotic control," in *CoRL*, 2023.
- [17] N. W. Alharthi and M. Brandão, "Physical and digital adversarial attacks on grasp quality networks," in *ICRA*, 2024.
- [18] J. Huang, H. J. Choi, and N. Figueroa, "Trade-off between robustness and rewards adversarial training for deep reinforcement learning under large perturbations," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8018–8025, 2023.
- [19] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *ICML*, 2018.
- [20] X. Zeng, C. Liu, Y.-S. Wang, W. Qiu, L. Xie, Y.-W. Tai, C.-K. Tang, and A. L. Yuille, "Adversarial attacks beyond the image space," in *CVPR*, 2019.
- [21] D. Wang, T. Jiang, J. Sun, W. Zhou, Z. Gong, X. Zhang, W. Yao, and X. Chen, "Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack," in *AAAI*, 2022.
- [22] J. Byun, S. Cho, M.-J. Kwon, H.-S. Kim, and C. Kim, "Improving the transferability of targeted adversarial examples through object-based diverse input," in *CVPR*, 2022.
- [23] N. Suryanto, Y. Kim, H. Kang, H. T. Larasati, Y. Yun, T.-T.-H. Le, H. Yang, S.-Y. Oh, and H. Kim, "Dta: Physical camouflage attacks using differentiable transformation network," in *CVPR*, 2022.
- [24] Y. Huang, Y. Dong, S. Ruan, X. Yang, H. Su, and X. Wei, "Towards transferable targeted 3d adversarial attack in the physical world," in *CVPR*, 2024.
- [25] L. Li, Q. Lian, and Y.-C. Chen, "Adv3d: Generating 3d adversarial examples for 3d object detection in driving scenarios with nerf," in *IROS*, 2024.
- [26] Y. Abeysirigoonawardena, K. Xie, C. Chen, S. H. Khorasgani, R. Chen, R. Wang, and F. Shkurti, "Generating transferable adversarial simulation scenarios for self-driving via neural rendering," in *CoRL*, 2023.
- [27] A. Chahe, C. Wang, A. Jeyapratap, K. Xu, and L. Zhou, "Dynamic adversarial attacks on autonomous driving systems," *arXiv preprint arXiv:2312.06701*, 2023.
- [28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.
- [29] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *NeurIPS*, 2020.
- [30] Z. Cheng, Z. Hu, Y. Liu, J. Li, H. Su, and X. Hu, "Full-distance evasion of pedestrian detectors in the physical world," in *NeurIPS*, 2024.
- [31] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *CVPR*, 2020.
- [32] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T. kai Chan, Y. Gao, X. Li, T. Mu, N. Xiao, A. Gurha, Z. Huang, R. Calandra, R. Chen, S. Luo, and H. Su, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *arXiv preprint arXiv:2410.00425*, 2024.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [35] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *ICAR*, 2015.